

RDX2

X

```
runmodTemplate *;@ModName@ <- function(Times, newParms,  
method="lsode",...){  
  Idots <- list(...); nstate <- @NSTATE@; nextra <-  
  @NEXTRA@;  
  Outnames <- @OUTNAMES@. ## Initialize parameters and state  
variables + Parm = initparms(newParms,@ParmVector@) + yinit <- Inity(Parms)  
  y <- yinit[[1]] + ydisc <- yinit[[2]]; ## Put ydisc at the bottom of Parm  
  $ ## initialize the events structure $ ev <- ConstructEvents(sort(Times)) 1 ## insert the  
events we already know about here $ @KnownEvents@ < ## update 'now' < now <-  
Times[1] + ## setup the matrix to receive the output - tmp <- matrix(nrow=0,  
ncol=1+nstate+nextra) ! ## run through the simulations.  
repeat {< if (ev$quit(now)) break % ys <- ev$doEvents(now,Parm,y,ev) + y <-  
ys[[1]] + ydisc <- ys[[2]] Parm[names(ydisc)] <- ydisc & nextint <-  
ev$getNextInterval(now) < tmp2 <- ode(y, nextint, "@DERIVS@",Parm, method=method,  
F dllname="@DLLNAME@", initfunc="@INITNAME@", nout=nstate, &  
outnames=Outnames,...) & y <- tmp2[nrow(tmp2),2:(nstate+1)]; now <-  
tmp2[nrow(tmp2),1] -> tmp <- rbind(tmp,tmp2) } # tmp <- tmp[!duplicated(tmp[,1])];  
1 structure(list(result=tmp[tmp[,1] %in% Times,], " parameters=Parm,  
# model="@ModName@", method=method,  
control=Idots), class="RDynOut") }p
```